



COMPUTER CENTER USER'S GUIDE

MANUAL S
SOS USER'S GUIDE .

1942

RECEIVED

RECEIVED

RECEIVED

RECEIVED

RECEIVED

RECEIVED

BUCKS COUNTY COMMUNITY COLLEGE
Newtown, Pennsylvania 18940

COMPUTER CENTER

USER'S GUIDE

Manual S
SOS USER'S GUIDE

first edition
(October 1977)

(C) Copyright 1977 by
Bucks County Community College
Computer Center
Newtown, Pennsylvania 18940

* all rights reserved *

No part of this publication may be
reproduced, stored or transmitted
in any form or by any means
without the prior written permission of
Bucks County Community College.

Introduction to the User's Guide

The BCCC USER'S GUIDE is intended to provide potential, new and current users with basic information concerning the purpose, facilities and capabilities of the BCCC Computer Center. It contains information about working and programming in the BCCC/CC environment.

The User's Guide is divided into several major sections which are published as separate documents. Thus, Users need to select and acquire only those sections required to satisfy their particular computing needs. The BCCC User's Guide consists of the following sections:

SECTION A -COMPUTER SERVICES AND USER'S GUIDE

This section outlines the services provided by the Computer Center Office, describes how the user can make use of these services, and provides a general guide to using the BCCC computer system in both time-sharing and batch processing modes.

Section A consists of the following chapters:

- Computer Center organization and structure,
- Computer Center services,
- Hardware - description and operating instructions,
- Software - commands, programs and features,
- the "BATCH" user - how to use the batch system,
- the "TERMINAL" user - how to utilize a terminal,
- various appendices,
- glossary of data processing terms,
- and
- an INDEX to help you easily locate topics or keywords.

SECTION B -BATCH PROGRAMMER'S GUIDE FOR PROGRAMMING LANGUAGES

This section outlines the procedures for running programs in the Batch operating System. This section alone will normally provide sufficient information for students enrolled in Computer Science courses.

Section B of the User's Guide consists of the following chapters:

- Project Programmer Numbers,
- Computer Facilities Locations and Scheduling,
- Reference Manuals and Materials,
- Compilers and Programming Languages,
- the "BATCH" user - how to use the batch system,
- various appendices,
- and
- an INDEX to help you to easily locate topics or keywords.

SECTION S -SOS TEXT EDITOR USER'S GUIDE

This section explains how to use the text editor language, SOS. This manual is written for either a beginning or experienced user starting to learn SOS. The manual explains and provides examples of all SOS commands and their uses.

Section S consists of the following chapters:

- How to create a brand new file,
- How to change and edit an existing file,
- How to end SOS and save your file,
- Advanced use of SOS,
- various appendices,

and
an INDEX to help you to easily locate topics or keywords.

Please note that the chapter name is printed as part of each page title to assist you in identifying the various sections of this guide. There is also a "reader comment form" on the last page of this guide.

1.0 INTRODUCTION TO SOS

The easiest way to create a file is by using a system program commonly called an TEXT-EDITOR. Editors, in addition to being able to create files, are capable of changing a file without requiring you to completely retype its contents.

SOS is the primary text editor in use on DEC-10'S and DEC-20'S. It is a line oriented editor as opposed to TECO which is character oriented. SOS is a mnemonic for "son of stopgap" and we have not been able to ascertain what that means, but we know SOS works well.

A file is a place in the DECsystem-10 where you are allowed to store information. The types of information users commonly store in files are:

1. Programs,
2. Data, and
3. Text.

When creating a file, you must assign it a name and extension for purposes of identification. This name and extension, coupled with the owner's identification called a directory name, distinguish your file from every other file in the system.

When you create a file, SOS automatically assigns each line a unique number. To reference that line again, all you have to do is include its line number in the SOS command you are giving.

Some of the functions the SOS commands perform are:

1. Printing lines (P command),
2. Deleting lines (D command),
3. Inserting lines (I command),
4. Replacing lines, i.e. Deleting and inserting combined in one command, (R command),
5. Finding parts of your file (F command),
6. Changing part of a line without completely retyping it (S command),
7. Copying lines (C command), and transferring lines (T command) within your file,
8. Ending editing and saving the file on disk storage (E command).

Remember, all SOS does is help you prepare a file so you can use it with another program. For instance, you may write a program, then enter it into a file using SOS. To run that program, you must give the execute command along with the name and extension of the file. Just entering the program into a file does not make the system run that program. If the program is to read

data, you may enter that data into a second file, again using SOS. The data is not used until you instruct the program to read it from the second file.

2.0 HOW TO CREATE A BRAND NEW FILE

2.1 STARTING SOS (.SOS)

To start SOS, LOGIN and type SOS, then press the return key. Your terminal leaves a blank line, then prints the word FILE:.

```
.SOS  
FILE:
```

SOS will store your program, data, or text in an area inside the computer called a DISK-FILE, each disk file has a name and an extension. The name contains up to six alphanumeric characters (letters and numbers) and the extension up to three. You must separate the name and extension by a period. Five examples of file names and extensions are: CALCS.FOR, SQUARE.ALG, PHONE.CBL, NIM.BAS, AND EXIT.TXT.

If your file is to contain programming language source code, tell the computer which programming language you are using by selecting one of the standard extensions listed below. If your file contains data or text, you may use any extension. For example, if you write a FORTRAN program, your extension should be .FOR; if you write a COBOL program, your extension should be .CBL. If you write a BASIC program, your extension should be .BAS. If you write a ALGOL program, your extension should be .ALG.

Choose a name and extension for your file (the example uses NUMBER.FOR). After your terminal prints the word FILE:, type the name and extension you have chosen, then press the return key. If you do not already have a file with this name and extension, your terminal prints the word INPUT:, repeats the name and extension of your file, and prints the number 00100. See the example below.

```
.SOS  
FILE: NUMBER.FOR  
INPUT: NUMBER.FOR  
00100
```

If you already have a file with this name and extension your terminal prints the word EDIT: (instead of INPUT:) and an asterisk (instead of the number 00100). The word EDIT indicates that you may now make changes to the existing file, instead of creating a new one. If you want to change your file, skip to the chapter on Making Changes to Your File; if you really want to enter a new file, follow the directions in the next section.

To create a new file after receiving the response EDIT:, first tell SOS to forget about changing the old file by typing the "END AND QUIT" command, EQ. Your terminal prints a period. Start over again by typing SOS, but this time choose a different name and extension. See the example below.

```
.SOS  
FILE: NUMBER.FOR  
EDIT: NUMBER.FOR  
*EQ
```

```
.SOS
FILE: NUMBRE.FOR
INPUT: NUNBRE.FOR
00100
```

You have now started SOS and are ready to enter the contents of your file.

2.2 ENTERING THE CONTENTS OF A FILE

To enter a program into a file all you have to do is type each line; then end it by pressing the return key. SOS prints the next line number; you type the next line; then press the return key. When you want to stop entering your file, press the escape key.

Type the short fortran program in the example below. After you finish typing the last line, proceed to the section entitled "ENDING SOS". Since nobody is a perfect typist, the system provides for methods for correcting typing errors; these are described later.

```
00100      TYPE 101
00200  101  FORMAT (' TYPE A NUMBER.')
```

ACCEPT 102,X

```
00400  102  FORMAT (F)
00500      TYPE 103,X
00600  103  FORMAT (' YOU TYPED ',F)
00700      END
```

2.2.1 LINE-NUMBERS - line numbers are used so you can easily reference each line in your file. Line numbers are not part of a program. In particular, the line numbers are not the label that marks the destination of any type of "go to" statement.

When you save your file with an E command, SOS saves the line numbers along with your file; when you give the ES command, SOS strips the line numbers off before saving the file. Some user programs may not recognize line numbers, so it is wise to use the ES command for files that will be read by other programs.

Line numbers consist of five digits followed by a tab. Thus, "column 1" of your file is located immediately following the tab at the end of the line number. This implies that column 7 is really the seventh character of the line. See the example below.

```
00600  103  FORMAT (' YOU TYPED ',F)
        ^="COLUMN 1"
        ^="COLUMN 7"
```

2.3 ERASING-CHARACTERS ON THE CURRENT LINE

To erase the last character that you typed, press the key labeled RUBOUT . Your terminal prints a backslash and the character that you have erased. Press the rubout key once for each successive character that you want to erase; the terminal prints each character as you erase it. When you continue typing, the terminal prints a second backslash to set off and enclose the erased characters. In the example, suppose that you just typed FORMST and want to correct it to be FORMAT.

```
00100 101 FORMST
```

To erase the t, press the rubout key; the terminal prints a backslash and the letter T

```
00100 101 FORMST\T
```

To erase the S, press the rubout key again; this time the terminal prints only the S.

```
00100 101 FORMST\TS
```

Next, type an A. The terminal prints a backslash just before the A to enclose the erased characters in backslashes.

```
00100 101 FORMST\TS\A
```

Finally, type the T and you are finished.

```
00100 101 FORMST\TS\AT
```

2.3.1 VIEWING THE LINE AFTER ERASING - to see the line after you have erased some characters, type a CTRL-R. Your terminal prints the line, but does not indicate the CTRL-R by a printing character. If you want to continue typing after the CTRL-R, do so immediately after the end of the line. You are still permitted to erase.

```
00100 101 FORMST\TS\AT
CTRL-R
101 FORMAT
```

2.3.2 ERASING AN ENTIRE LINE - to erase an entire line, type a CTRL-U. Retype its contents on the next line. SOS does not retype the line number.

```
00100 101 FORMST^U
101 FORMAT
```

To type a control character, press the key labeled CTRL, and hold it down while you type the character. For a CTRL-U you will press the CTRL key and type a U. The terminal prints most control characters (remember, CTRL-R is not printed) as an up-arrow (^) followed by the character. Thus, CTRL-U (discussed in the next section) appears on your terminal as: ^U. The RUBOUT key, CTRL-R, and CTRL-U work only for the line you are currently typing. Once you press the return key, you must use other SOS commands (D,I,R,S OR P) to change or print the line.

2.4 STARTING THE FILE OVER AGAIN

If you want to start the file over again, press the key labeled ESCAPE ; your terminal prints a dollar sign, then an asterisk. Next, type EQ and press the return key; the terminal prints a period. You can now start all over by typing SOS.

```
00100 101 FORMST$  
*EQ
```

2.4.1 ENDING SOS - after you have finished typing the program, press the ESCAPE-KEY. The terminal prints a dollar sign, and on the next line, an asterisk.

```
00100 TYPE 101  
00200 101 FORMAT (' TYPE A NUMBER')  
00300 ACCEPT 102,X  
00400 102 FORMAT (F)  
00500 TYPE 103,X  
00600 103 FORMAT (' YOU TYPED ',F)  
00700 END  
00800 $  
*
```

To save the file, type E and press the return key. SOS leaves a blank line, prints the name and extension of your file, and then prints a period on the next line.

```
00100 TYPE 101  
00200 101 FORMAT (' TYPE A NUMBER.')
```

```
00300 ACCEPT 102,X  
00400 102 FORMAT (F)  
00500 TYPE 103,X  
00600 103 FORMAT (' YOU TYPED ',F)  
00700 END  
00800 $  
*E  
[DSKC:NUMBER.FOR]
```

3.0 HOW TO CHANGE AND EDIT AN EXISTING FILE

Now you can make changes to the file you have previously entered. Start SOS by typing SOS. After the terminal prints FILE:, type the name of your file. This time, SOS responds by printing EDIT: the name of the file, and an asterisk. Refer to the example below.

```
.SOS
FILE: NUMBER.FOR
EDIT: NUMBER.FOR
*
```

3.1 PRINTING LINES (PRINT-COMMAND)

to print the contents of the file, type P and press the return key. SOS prints up to 16 lines of your program at one time. In the example, the user's program has only seven lines, so the P command prints the entire program.

```
*P
00100          TYPE 101
00200    101    FORMAT (' TYPE A NUMBER.')
```

```
00300          ACCEPT 102,X
00400    102    FORMAT (F)
00500          TYPE 103
00600    103    FORMAT (' YOU TYPED ',F)
00700          END
*
```

After printing the lines, your current place in the file is at the last line that SOS printed; in the example that would be line 700. Since your position in the file is moved, if you give a second P command, SOS reprints line 700 and continues with the next 15 lines of your file.

3.1.1 PRINTING ONE LINE - if you want to print just one line, type a P, the line number of the line, and press the return key.

```
*P500
00500          TYPE 103
*
```

Whenever a line number has leading zeroes (such as 00500), you do not have to type them when you give a command which has a line-number argument.

3.1.2 PRINTING A GROUP OF LINES - to print a group of lines, type a P, the line number of the first line, a colon, the line number of the line, and press the return key.

```
*P500:700
00500          TYPE 103
00600    103    FORMAT (' YOU TYPED ',F)
00700          END
*
```

3.1.3 PRINTING THE FIRST LINE OF YOUR FILE - to print the first line of your file, type a P, an up-arrow, and press the return key.

```
*P~  
00100          TYPE 101  
*
```

3.1.4 PRINTING THE LAST LINE OF YOUR FILE - to print the last line of your file, type a P, an asterisk, and press the return key.

```
*P*  
00700          END  
*
```

3.1.5 PRINTING THE ENTIRE FILE - to print the entire file, type a P, up-arrow, a colon, and an asterisk, then press the return key.

```
*P~:*  
00100          TYPE 101  
00200    101    FORMAT (' TYPE A NUMBER,')  
00300          ACCEPT 102, X  
00400    102    FORMAT (F)  
00500          TYPE 103  
00600    103    FORMAT (' YOU TYPED ',F)  
00700          END  
*  
*
```

3.1.6 PRINTING THE CURRENT LINE - whenever you print a line or lines, SOS "remembers" the last line it printed. In fact, after every legal SOS command, SOS "remembers" the last line as its current place. To print your current place, type a P, a period, and press the return key.

```
*P.  
00700          END  
*
```

3.1.7 PRINTING THE PREVIOUS LINE - \$ - to print the previous line, press the ESCAPE-KEY.

```
*$  
00600    103    FORMAT (' YOU TYPED ',F)  
*
```

3.1.8 PRINTING THE NEXT LINE - <LF> - to print the next line, press the linefeed (or <LF>) key; no character is printed to show you pressed this key. the symbol <LF> shows where you would press the linefeed (or <lf>) key.

```
*<LF>  
00700          END  
*
```

3.1.9 SUMMARY OF PRINT COMMANDS -

*P	prints the next 16 lines, if possible
*P500	prints line 500
*P500:700	prints lines 500 through 700
*P	prints the first line in your file
P	prints the last line in your file
P:	prints the entire file
*P.	prints the current line
*\$	(pressing the escape key) prints the previous line
*<LF>	(pressing the linefeed key) prints the next line

3.2 DELETING LINES (DELETE-COMMAND)

To delete a line, type a D and the appropriate line number; then press the return key. SOS prints a message indicating the line you deleted.

```
*D100
1 LINES (00100/1)  DELETED
*
```

To delete a group of lines, type a D, the number of the first line, a colon, and the number of the last line. Finally, press the return key. SOS prints a message indicating the lines you deleted.

```
*D600:700
2 LINES (00600/1:00700)  DELETED
*
```

If you should mistakenly delete some lines, you may retrieve them by ending SOS with the EQ command. The disadvantage of giving the EQ command is that you also lose everything else that you did to the file since you save the last .SOS command.

Your place after a D command is at the last line that you deleted.

3.3 INSERTING LINES (INSERT-COMMAND)

To insert a line, type I, the number of the line you want to insert, and press the return key. SOS types the line number and waits for you to finish typing the line.

```
*I100
00100          TYPE 101
*
```

If you already have a line 100 in your file, SOS will choose either line 200 (if it does not already exist), or line 150. If you then try to insert line 150, SOS will choose a line halfway between 150 and 200, which is 175.

When you are at the end of a file, or at a place where line numbers differ by more than 100, SOS allows you to insert more than one line. That is, after you type a line, SOS prints another line number. You type the contents of the line and press the return key. If a conflict arises (such as trying to

insert an existing line), then SOS stops inserting lines. If you want to stop this automatic procedure of letting you insert line after line, press the escape key to get the asterisk. The example below inserts the last two lines of the program (slightly changing the former line 600).

```
*I600
00600 103      FORMAT (' YOU TYPED THE NUMBER ',F)
00700      END
00800  $
*
```

3.4 REPLACING LINES (REPLACE-COMMAND)

To delete a line and then insert one directly in its place, type R, the number of the line, and press the return key. SOS prints the line number and waits for you to type the line. Then, after you finish, SOS prints the message just as if you gave a D command.

```
*R200
00200 101      FORMAT (' PLEASE TYPE A NUMBER. ')
1 LINES (00200/1) DELETED
*
```

Using the R command is exactly like giving a D command directly followed by an I command. First, SOS deletes the line or lines; second, SOS enters input mode. This means that after an r command SOS stays in input mode when you are located at the end of a file or between line numbers that differ by more than 100. To obtain the asterisk, simply press the esc key to return to command mode.

3.5 NUMBER-COMMAND

You will find it much easier to type line numbers if they are even multiples of 10 or 100. The N command renumbers the line numbers in your file.

3.5.1 RENUMBERING, STARTING WITH 100 AND ADDING 100 (NUMBER-COMMAND) - to renumber your file, starting with 100 and adding 100 to get each sequential line number, type N and press the return key. After SOS renumbers your file, it leaves you positioned at the end of the file.

The example below shows the user enter three lines - lines 100, 200, and 300. Next, the user enters lines 150 and 250 with two separate I commands. The N command renumbers the file: line 100 is left alone; line 150 becomes the new line 200; line 200 becomes the new line 300; line 250 becomes the new line 400; line 300 becomes the new line 500. After the N command, the user gives the P. Command to show his place is at the end of the file. Lastly, he prints out the entire file.

```
.SOS
FILE: TEST.TXT
INPUT:TEST.TXT
00100  A. RESTON
00200  T. SIMOLE
00300  L. DELANEY
```

```
00400  $
*I150
00150  S. PATRIACH
*I250
00250  E. WHITE
*N
*P.
00500  L. DELANEY
*P^:*
00100  A. RESTON
00200  S. PATRIACH
00300  T. SIMOLE
00400  E. WHITE
00500  L. DELANEY
*
```

When you renumber your file, the lines are kept in exactly the same order.

3.5.2 RENUMBERING, USING A NUMBER OTHER THAN 100 - if you want to use some number other than 100, type N, the number you want to use, and press the return key. The example uses the same file as in the example in the previous section.

```
*P^:*
00100  A. RESTON
00200  S. PATRIARCH
00300  T. SIMOLE
00400  E. WHITE
00500  L. DELANEY
*N10
*P^:*
00010  A. RESTON
00020  S. PATRIARCH
00030  T. SIMOLE
00040  E. WHITE
00050  L. DELANEY
*
```

3.5.3 WRAP-AROUND - if SOS ever prints the message WRAP-AROUND, it has exceeded its maximum line number of 99999 and started at zero again. The lines of your file are in the proper order, but the line numbers are not in sequential order. To fix the line numbers, give the N command with a smaller number than you have previously given.

3.6 FINDING PARTS OF YOUR FILE - (FIND-COMMAND)

To find a string of characters within your file, type an F, type the characters you want to find, press the ESC key, and then press the return key. In the example, use the file NUMBER.FOR and search for the characters ACCEPT.

.SOS


```
FILE: NUMBER.FOR
EDIT: NUMBER.FOR
*FACCEPT$
00400          ACCEPT 102,X
*
```

After searching for a string, you are placed at the line where the characters were found. Thus, a F Command reprints the line containing the string.

```
*P.
00400          ACCEPT 102,X
*
```

This form of the F command finds the next occurrence of the string, from the current position through the end of the file.

3.6.1 SOS CANNOT FIND THE STRING - SEARCH-FAILS - if we tried to search for the characters hello, SOS would not find them and print the message search fails. In this case, your place is not changed.

```
*FHELLO$
SEARCH FAILS
*P.
00400          ACCEPT 102,X
*
```

3.6.2 FINDING A STRING LOCATED ANYWHERE IN THE FILE - to find the first occurrence of a set of characters starting at the beginning of the file and continuing to the end of the file, type F, the characters you are looking for, press the ESC key, type an up-arrow, a colon, and an asterisk, then press the return key.

```
*F101$^:*
00100          TYPE 101
*
```

If you want to continue looking for the string, all you have to type is F. SOS remembers the string for which you last looked.

```
*F
00200 101      FORMAT (' TYPE A NUMBER.')
```

3.7 CHANGING A LINE WITHOUT RETYPING IT - (SUBSTITUTE-COMMAND)

to change the contents of a line without completely retyping it, use the S command. For example, the previous contents of line 200 in the program were:

```
00200 101      FORMAT (' TYPE A NUMBER.')
```

Using the REPLACE command you changed it to:

```
00200 101      FORMAT (' PLEASE TYPE A NUMBER.')
```

By completely retyping the line. Now, add an exclamation point after the word PLEASE, but this time you will not have to retype the entire line. You will search for the word PLEASE and substitute in its place the word PLEASE!.

To give this command, type an S, type the word PLEASE, and press the ESC key. (So far, this command tells SOS to look for the word PLEASE.)

```
*SPLEASE$
```

Continuing on the same line, type PLEASE! and press the ESC key. (Now, the command tells SOS to look for PLEASE and replace it with PLEASE!.)

```
*SPLEASE$PLEASE!$
```

Again on the same line, type the line number 200 and then press the return key. (The 200 tells SOS to perform the substitution only on line 200.) After SOS performs the substitution, it prints the new line.

```
*SPLEASE$PLEASE!$200
00200 101 FORMAT (' PLEASE! TYPE A NUMBER.')
```

3.7.1 SOS CANNOT MAKE THE SUBSTITUTION - SEARCH-FAILS - sometimes, you might type some characters that are not in the line; in that case, SOS prints the message search fails and leaves you at your current location. The next example shows what happens if you try to change the word TYPE to choose, but misspell TYPE as TIPE.

```
*STIPE$CHOOSE$200
SEARCH FAILS
*
```

3.7.2 MAKING SUBSTITUTIONS THROUGHOUT YOUR FILE - you can easily make substitutions throughout your file. Let's print the entire program and then search for the variable X and replace it with the new variable called NUMBER. To print your entire program, type P, an up-arrow, a colon, and an asterisk, and then press the return key.

```
*P~!*
00100 TYPE 101
00200 101 FORMAT (' PLEASE TYPE A NUMBER.')
```

To search for the X and substitute the variable NUMBER for it, type S, type an X and press the ESC key (this tells SOS to look for an X.)

```
*SX$
```

On the same line, type the word number and press the ESC key (this tells SOS to replace the X with number.)

```
*SX$NUMBER$
```

Finally, type an up-arrow, a colon, and an asterisk; then press the return key. (This tells SOS to perform the substitution over the entire program.) SOS prints each line where it performs the substitution.

```
*SX$NUMBER$:*
00300 ACCEPT 102,NUMBER
```

```
00400          TYPE 103,NUMBER
*
```

3.7.3 CHOOSING THE PROPER SEARCH STRING - searching for the X was easy because X was used for only one thing in the entire program. Since the variable is now an integer, you must change the F inside the parenthesis of the format statements to an I. You do not want to change the F in the word format to an I. This means you have to be a little tricky in choosing a unique set of characters to search for and replace.

Notice that each of the F's you want to change to I's just happens to be followed by a right parenthesis-- there is your solution. search for an F followed by a right parenthesis and substitute an I followed by a right parenthesis; this S command will work over the entire program. The correct S command is:

```
*SF)$I):*
00300 102   FORMAT (I)
00600 103   FORMAT (' YOU TYPED THE NUMBER ',I)
*
```

In order to use the S command most effectively, you have to notice little tricks such as you used above. If you had just searched for the F and not bothered to use the right parenthesis, SOS would have also changed the F in the word format to an I.

```
*SF$I$:*
00300 102   IORMAT (I)
00600 103   IORMAT (' YOU TYPED THE NUMBER ',I)
*
```

3.8 COPYING LINES WITHIN YOUR PROGRAM COPY-COMMAND - TRANSFER-COMMAND

To copy a range of lines from one place in your file to another, type a C and the line number where you want the copied lines to start; leave a comma; then type the range of lines to be copied and press the return key.

For example, suppose you have the file:

```
*P~:*
00100 JUMP
00200 JUMPGE
00300 JUMPL
00400 JUMPA
00500 JUMPE
00600 JUMPG
*
```

and want to put the lines in alphabetical order. The following C command copies lines 400 through 600 to new lines immediately after line 100. Note that since line 100 already exists, SOS inserts the lines immediately after that line. The message incl=00020 tells you that SOS uses an increment of 20 when selecting new line numbers.

```
*C100,400:600
INC1=00020
*P~:*
```

```
00100    JUMP
00120    JUMPA
00140    JUMPE
00160    JUMPG
00200    JUMPGE
00300    JUMPL
00400    JUMPA
00500    JUMPE
00600    JUMPG
*
```

To avoid duplication, you should delete lines 400 through 600. However, to do this automatically, use the TRANSFER command.

Start with the same file as in the previous example; then give the following T command:

```
*T100,400:600
INC1=00020
*P~:*
00100    JUMP
00120    JUMPA
00140    JUMPE
00160    JUMPG
00200    JUMPGE
00300    JUMPL
*
```

If you set the message " INC1=ORDER ", then SOS was not able to choose an increment small enough to fit the lines between the destination line and the next line in the file. However, SOS still copies the lines, but the line numbers are out of sequential order. To preserve the integrity of your file, you must immediately give an N command to correctly renumber your file.

In review:

- the C command duplicates the source lines starting at a particular place in the file.
- the T command duplicates the lines and, in addition, deletes them from the source.
- upon setting the message: INC1=ORDER, always give an N command.

4.0 HOW TO END SOS AND SAVE YOUR FILE

To end SOS and save your file, use one of the following commands:

1. The E command ends SOS, saving the file along with its line numbers.
2. The ES command strips the line numbers (i.e., unsequences the file), saves the file, and ends SOS.
3. The G command ends SOS, saves the file along with its line numbers, and then executes your last compile, load, execute, or debug command.

4. The EQ command ends SOS, but does not save the changes to the file.

Also, the W command allows you to save your file and continue editing without ending SOS.

4.1 SAVING THE FILE WITH LINE NUMBERS - (END-COMMAND)

To end SOS, saving both the file and the line numbers, type E and press the return key. SOS prints the name of the file you have saved and leaves you at command level (signified by the printing of the period).

```
*E
[DSKC:STATS.DATE[27,4020]]
.
```

Some programs do not operate properly if you give them an input file that has line numbers. If you find a malfunction of this sort, run SOS, but use the ES command (instead of Just E) to remove the line numbers.

SOS automatically renames the original file to FILE.QXX. For example, the original STATS.DAT file would be called STATS.QAT after editing, thus providing a convenient way to check changes. You may delete a .QXX file with a system delete command, but should only do so after you are sure that you have made the proper changes to the file.

4.2 SAVING THE FILE WITHOUT LINE NUMBERS END-AND-STRIP-COMMAND

To end SOS and strip off the line numbers, type ES and press the return key. SOS removes the line numbers, prints the name of the file, and then leaves you at command level.

```
*ES
[DSKC:STATS.DATE[27,4020]]
.
```

The ES command is useful when you are preparing a file that will be read by a program that does not recognize line numbers left in the file by the E command. The ES command also creates a backup file with the extension .QXX.

4.3 SAVING THE FILE AND EXECUTING THE LAST COMMAND GO-COMMAND

The G command can save you some typing. Suppose you have previously given the command:

```
.EXECUTE CUBIT.FOR
```

and have since given no other compile, log, execute, or debug command. (These commands are referred to as the compile class commands.) if you then give the G command to end SOS, SOS will save the file and also execute the execute command. See the example following:

```
.SOS
FILE: CUBIT.FOR
EDIT: CUBIT.FOR
```

```
*S $)$400
00400 102 FORMAT (' TYPE THE NUMBER OF OCCURRENCES: ')
*G
EDSKC:CUBIT.FORC27,402011

FORTRAN: CUBIT
MAIN.
LINK: LOADING
[LINKSCT CUBIT EXECUTION]
TYPE THE NUMBER OF OCCURRENCES: 34
THANK-YOU

END OF EXECUTION
CPU TIME: 0.08 ELAPSED TIME: 3:15.13
EXIT
```

The G command also creates a backup file with the extension .QXX.

4.4 ENDING WITHOUT SAVING YOUR WORK - (END-AND-QUIT-COMMAND)

To end SOS without saving the work you did, type EQ and press the return key.

```
*EQ
```

The EQ command does not save the work you did since you save the last .SOS command. This means that if you created a file, it is now gone, or if you edited a file, this last set of edits is gone. Since SOS does not have to do any work, the EQ command operates very quickly.

The EQ command is useful in the following circumstances:

1. When you have used SOS to merely read through a file, there is no reason to resave it because you have not made any changes. In this case, SOS leaves the file untouched.
2. After you have given a command that makes your file useless, you can give the EQ command to eliminate all you have done to the file. For example, suppose you save a D command by mistake. If it is worth losing the edits you have just made you can give the EQ command and recover the lines you lost. Also, if you want to try a particular S command, you can start SOS (or give a W command) and try the S command. Then if the command does not work the desired way, give an EQ command and you will have the original file.

The EQ command does not create a backup file.

4.5 SAVING THE FILE WITHOUT ENDING SOS - (WORLD-COMMAND)

the W command saves your file, but lets you continue editing without stopping SOS. It is good practice to give a W command every 15 minutes or so. That way you are sure of retaining your work no matter what happens. Many SOS users also agree on another basic rule of editing - never walk away from your terminal without giving a W (or an E) command to protect yourself from a

naive person who might type an EQ command.

*W

[DSKC:STATS.DAT[27,4020]]

*

The W command creates a BACKUP-FILE with the extension .QXX.

5.0 ADVANCED USE OF SOS

5.1 EXPLANATION OF SOS FILE CONCEPTS

5.1.1 STANDARD TEXT FILES - SOS utilizes a Standard Text File, containing Standard Text Lines and Standard Page Marks. Standard Text Line format is a line number composed of a string of five seven-bit ASCII digits, left-justified in a single word with a 1 in bit 35. This line number is followed by a TAB and the text (147 or fewer ASCII characters, not including the carriage return/linefeed). Each disk file word is left-justified and consists of five ASCII characters and a 0 in bit 35. The first character which follows the line number is a TAB and the last two characters are (CR)(LF). The characters NULL and DELETE may not appear in the text. If necessary, the last disk file word for the line is filled with NULLs to make a complete word.

A Standard Page Mark is a word containing 5 ASCII spaces, left-justified in a word with a 1 in bit 35, followed by a word containing the characters (CR),(CR),(FF),NULL,NULL, with a 0 in bit 35.

Lines are never broken across physical disk block boundaries, and all unused words in a disk block are 0.

If the specified file was created by using an editor that does not assign line numbers, SOS will assign a number to each line as it reads the file, beginning with 100 and incrementing by 100.

5.1.2 LINE-NUMBERS , PAGES - Throughout the earlier sections of this guide, it was assumed that a file was entirely contained in one continuous group. This is a feasible way to structure a file; however, when a file is large, it is advantageous to divide it into pages. For every command that includes a line number specification, there is also a page qualification. Thus, a single line may be specified by both the line number and page number as follows:

<LINE NUMBER>/<PAGE NUMBER>

The line number must be in the range 0 to 99999; the first page is 1, and there is essentially no upper limit. A range has the form:

<LINE NUMBER>/<PAGE NUMBER>:<LINE NUMBER>/<PAGE NUMBER>

A range includes all text up to and including the endpoints. Instead of a number, the symbol "." may be used to denote either the current line or the current page, depending on whether it appears before or after the "/". Therefore, "100/." means line number 100 on the current page and ". /2" means the line on page which has the same number as the current line number. The current line and page are determined by the last command which was executed (or, generally, by the last line on which an operation was performed and the page on which that line appears).

The symbol "*" may be used to specify the last line on the page, but it may not specify the last page.

Line and page specifications may be further abbreviated as follows:

Abbreviated Specification	Full Specification
---------------------------	--------------------

100	100/.
100:500	100/.:500/.
100/2:500	100/2:500/2
/2	0/2:* /2
/2:/5	0/2:* /5
0/1:.	0/1:./1

Addition and subtraction can also be used in line and page specifications. While the arithmetic is straight forward for pages, it is not for lines because of varying increments and prior editing changes. A count either goes forward or backward from the indicated line to include the number of lines specified. Another form of the range specification extends the ability to use the counting technique. The starting line is specified in one of the above ways and is followed by an "!" and the number of lines desired.

Line specifications using arithmetic cannot cross page boundaries, except those with the "!" which may cross page boundaries if necessary.

Special Specification	Equivalent Specification
100+2	the second line after line 100
100!3	line 100 and the two subsequent lines
/5-3	/2 or page 2
/.+2	the second page after the current page
/.-1	the previous page
.-1!+.1	three lines, including the current line
*+2	*
0-5	0
*!2	* /.:0+1 /+.1

5.2 EDIT-MODE-COMMANDS

5.2.1 The MARK-COMMAND - "M" - The M command will insert a page mark into the text, immediately before the line specified. Note that page mark 1 is always assumed when input is begun.

*M<LINE NUMBER>/<PAGE NUMBER><CR>

If the line that is specified does not exist, the page mark will be inserted immediately after the line with the next lower number on that page. If the line specified is the first on that page, the new page mark will be placed immediately after the already-existing page mark; therefore, the page that

already existed will not have any accompanying text. If the page specified in the M command does not exist, the error message "%NO SUCH PAGE" will be printed and no page mark will be inserted. Note that the page mark always precedes the related text.

Inserting a page mark causes renumbering of all subsequent pages. The current line is set to zero; the current page is considered the one just created. When the user is inserting text, the command

*M99999<CR>

will place a page mark at the end of the current work, and the next insertion will go on the next page.

5.2.2 The DELETE-COMMAND - "D" - The Delete command will delete all the lines of a specified range:

*D<range><CR>

If there are no lines in the specified range, SOS will respond with the error message "%NO SUCH LINES(S)".

Lines in a range are specified in the usual manner; exceptions are noted below:

1. The Delete command does not allow a page specification for the second line number of a range. All deletions extend, therefore, over one page only, even if the "!" is used.

*D100/2:200/3<CR>

This is an illegal command; SOS will respond with an appropriate error message.

2. The Delete command will delete a page mark, not the contents of the given page:

*D/5<CR>

Delete page mark 5 (not the contents of page 5)

The numbers of the succeeding page marks are decreased by one.

Deleting a page mark may result in the creation of a page with line numbers that are out of sequence. If so, SOS will print the message "%OUT OF ORDER". In such a case, it is necessary to renumber the page immediately.

Page mark 1, or any page mark which does not exist, may not be deleted. If deletion is attempted, the error message "%NO SUCH PAGE" will be printed.

All lines on a page may be deleted as follows:

*D/:99999<CR>

Delete all lines on page 2.

The current line and page will be set to the last line deleted on the specified page.

5.2.3 The INSERT-COMMAND - "I" - The Insert command, which is used to insert new lines into a file, accepts a single line specifier (line number/page number) as its first argument and begins inserting at that line. The Insert command may also be used to change the current default increment between lines, in which case the increment is preceded by a comma.

*I<line>I[,<increment>]<CR>

or

*I<line>I[,<increment>]<CR> Thus,

*I100/2,50<CR>

Insert lines starting with line 100 on page 2 and set the current increment to 50.

*I100/2;50<CR>

Insert lines starting with line 100 on page 2. Use the increment of 50. NOTE: The semi-colon indicates that 50 is a temporary increment to be used only while this command is in effect. The current increment will remain unchanged and will be used again for the next command.

*I100/2<CR>

Insert lines starting with line 100 on page 2 using the current increment (since no increment is specified). NOTE: At the beginning of edit mode, the current increment is 100.

After a specified line is actually inserted, SOS will add that line number to the correct increment to obtain the number of the next line to be inserted.

If there already is a line with the same number as the specified line number, the line number of the existing line is added to the correct increment and insertion begins there. If there already is a line located between the two line numbers (or at the second line number), SOS will select a line number that is halfway between the indicated line number and the next line in the file. For the next insertions, if a line already exists at the location determined by the increment, or if it is necessary to pass over a line which already exists, then insertion mode will terminate and the user will be returned to command level.

In the following example, lines 100 and 200 already exist; the current increment is 100:

*I100<CR>

00150 THIS IS A LINE<CR> SOS selects line 150 to begin insertion. The user is returned to command level since line 200 already exists and must be passed over (200 is between 150 and 250).

If a line insertion is terminated with an \$ (ESCAPE), the text preceding the \$ is inserted, and the user is returned to command mode. If nothing has been inserted and \$ is used, an empty line will not be inserted, although the user will be returned to command level.

The current line number and page become the last line actually inserted, not the empty line terminated with \$. In addition, after \$ is typed, SOS will remember the line number and page of the next line to be inserted; these will be used if the I command is given later with no argument. The increment will be the same as the increment used in the most recent complete command, including those commands with the temporary (";") construction.

```
*I100;100<CR>
00100 THIS IS LINE 1.<CR>
00200 THIS IS LINE 2.<CR>
00300 $
*P100<CR>
00100 THIS IS LINE 1.
*I<CR>
00300 THIS IS LINE 3.<CR>
00400      .
          .
          .
```

If a page which does not exist is specified, SOS will respond with "%NO SUCH PAGE".

If an attempt is made to insert a line which contains more than 147 characters (not including the return/linefeed at the end), SOS will print "%LINE TOO LONG". The line will not be inserted, and SOS will return to command mode.

If the next line to be inserted has a line number greater than 99999, insertion will terminate and the user will be in command mode.

An increment of zero will result in the error message "%ILLEGAL COMMAND".

5.2.4 The REPLACE-COMMAND - "R" - The Replace command is the same as the Delete command followed by the Insert command. Replace accepts a range specifier and an optional increment, which can be either temporary ("\$") or an actual ("") update of the current increment.

```
*R<range>[; <increment>]<CR>
      or
*R<range>[, <increment>]<CR>
```

The deletion is performed on the specified range, and insertion begins with the first line indicated by that range. If the first line is not stated explicitly, the line number of the first line deleted will be used as the starting point for insertion.

```
*R/2,50<CR>
```

Replace all lines on page 2 and insert new lines, beginning with the first line deleted, using an increment of 50.

Although the Delete command will delete a page mark and not the contents of that page, the Replace command will delete all of the lines on the specified page and then allow insertion.

```
*D/2<CR>
```

Delete page mark 2

```
*R/2<CR>
```

Delete all the lines on page 2

Insertion may be terminated by depressing \$ after SOS has printed the next line number. As in the Insert command, if \$ is used after some of the text has been typed, that text will be inserted and command mode will be reinstated.

5.2.5 The FIND-COMMAND - "F" - The Find command is used to locate a given string of text within a stated range (note that the options are discussed separately below).

*F[<STRING>] \$ [<RANGE>][,<OPTIONS>]]<CR>

To find the first occurrence of a given text on a specific page, simply precede the page number with a "/" as follows:

*FLINE \$ /2<CR>

Find and print out the first line on page 2 containing "LINE".

To locate the same string specified in the last Find command, only use the range argument.

*F \$ /3<CR>

Find and print out the first line on page 3 containing "LINE" (from the previous Find command).

If no previous Find command was issued, SOS will print "%NO STRING GIVEN" since no string had been specified.

If the range is omitted, then SOS will begin its search at the line following the current line and continue through the end of the file.

*FALWAYS \$ <CR>

Find the first occurrence of "ALWAYS" beginning with the line following the current line through the end of the file.

To indicate a range beginning with the next line through a specified location, it is necessary only to specify the second half of the range.

*FALWAYS \$:/5<CR>

Find the first occurrence of "ALWAYS" in the range from the next line through the end of page 5 (assuming the current position is before page 5).

Note that if this particular range specification were used in any other command (except Substitute), SOS would print the message "%ILLEGAL COMMAND".

When the Find command is given repeatedly, it is not necessary to specify the text and range when referring to the same text and range given in the last complete Find command.

*F<CR>

Continue to find the text (specified previously). The search begins with the line after the last line that was found; the end of the range is the endpoint specified in the last complete Find command.

*F \$ <CR>

Continue to find the text (specified previously) through the end of the file (since no range had been specified).

The current line will be set to the last one found. If the error message "%SEARCH FAILS" is printed, the string cannot be located in the remaining portion of the range. In this instance the value of the current location will be unchanged.

It is possible to use an extended form of the Find command to find up to six

strings of text. The carriage return separates the text strings:

```
*F<string1><CR>          Search for any occurrence of any of
<string2><CR>          the strings (totalling fewer than
.                        200 characters) in the range
.                        specified.
.
<stringn> $ <range><CR>
```

The complete Find command format including options, is as follows:

```
*F[[<STRING>] $ [[<RANGE>][,A ,N][,E][,<NUMBER>]]<CR>
```

Find command options are defined below:

A Enter intraline edit mode automatically when the string is found. This moves the intraline pointer to the first character of the string. If intraline editing is complete and "*F<CR>" is issued, the A option will still be in effect. (This is the same for the N and E options discussed below).

N Find just the line numbers of the locations of a given string. N follows the range specification and cannot be used with the A option. N will still be in effect if an "*F<CR>" is issued.

E Do not treat upper and lower case letters as being equivalent. E follows the range, or A or N if either is present. E will still be in effect if an "*F(CR)" is issued.

<number> "<number>" specifies the number of lines to be found - 99999 usually will find them all. This option can be used instead of repeating "*F(CR)" until the string no longer can be found within the range specified in the first complete Find command.

5.2.6 The COPY-COMMAND - "C" -

The Copy command will insert a copy of existing text at a given location. Its format is:

```
*C(DESTINATION LINE NUMBER),(SOURCE RANGE)[,(INC1)[,(INC2)]]<CR>
```

The Copy command acts as if an Insert command had been done for a given location, and then as if all of the lines specified by the source range had been typed in, using the specified increment between destination line numbers. If the increment is too large, SOS will print "%OUT OF ORDER" or "%WRAP AROUND", and then choose a smaller increment, which is indicated as follows:

```
INC1=<number>
```

If, for some reason, SOS cannot choose a small enough increment, it will print

```
INC1=ORDER or INC1=WAR
```

The specified increment will be used, and SOS will copy as many lines as possible into the available space.

Since all the text to be copied must be contained in core at one time, copying large blocks of text may result in the error message "%INSUFFICIENT CORE AVAILABLE". The only solution is to copy several smaller blocks of text.

If the source lines contain page marks, the renumbering of lines at the destination using the first increment will cease when the first page mark is reached; then all lines between the first and last page marks will be inserted with original numbers. Lines after the last page mark will be inserted with the original numbers, unless a second increment (inc2) for the destination line numbers is specified in the command.

If the second increment is too large, SOS will print one of the following error messages: "%OUT-OF-ORDER" or "%WRAP-AROUND".

If the second increment is not specified and renumbering is not done, SOS will print "%OUT OF ORDER".

In both cases, SOS will try to use a different increment, as indicated by
INC2=NUMBER

If SOS cannot find a suitable increment, it will print
INC2=ORDER

SOS then will use the specified increment (or the original line numbers if a second increment had not been stated in the command), and will insert as many lines as possible into the remaining available space.

The Copy command will enable the user to copy lines from a file other than the one presently being edited. If the line numbers of this other file are unknown, then the user will first type:

*C(DESTINATION LINE NUMBER)=(SOURCE FILENAME)/S<CR>

This will cause SOS to respond with an asterisk (*). "/S" places the indicated source file in read-only mode. The desired lines are found by using the Print, List, or Find commands. After the lines have been found, type

*E(CR)

SOS will reply
SOURCE LINES=

At this point, the rest of the Copy command should be typed.
<SOURCE RANGE>[,<INC1>[,<INC2>]]<CR>

The current line will be set to the last line copied.

5.2.7 The JOIN-COMMAND - "J" - The Join command Joins two consecutive lines by deleting the carriage return/linefeed sequence at the end of one line and the line number and TAB of the next line.

*J<line><CR>

<line> is the first of the pair of lines to be joined. The line number of the new line that is created will be the same as this first line.

If there are no errors, the current line is the line that is created.

The lines will not be joined if SOS prints the error message "%LINE TOO LONG".

If the line specified as the argument is the last line on the page, SOS will give the error message "%NO NEXT LINE".

5.2.8 The NUMBER-COMMAND - "N" - After an "%OUT-OF-ORDER" message (or at any other time), it is possible to renumber the file, or portions of it, by using the Number command:

*N(INCREMENT)[,(RANGE)[,(STARTING NUMBER)]]<CR>

The first argument is the increment to be used in the renumbered file. The first line that is renumbered is given this number, unless the third argument is specified. Each succeeding line will be given a number that is the sum of the increment and the previously assigned line number.

If renumbering crosses a page mark, the first line on the new page will be given the "starting number" - i.e., either the third argument or the first argument (if the third is omitted). For example,

BEFORE RENUMBERING:

PAGE 2 115 126 137 150

PAGE 3 900 911 952 999

*N10,/2:/3<CR>

AFTER RENUMBERING:

PAGE 2 10 20 30 40

PAGE 3 10 20 30 40

If the second argument is absent (indicating the range to be renumbered), the entire file will be renumbered. If there are no lines in the specified range, SOS will print the error message "%NO SUCH LINE(S)".

The Number command is useful in renumbering a page before the page mark is deleted in order to avoid an "%OUT OF ORDER" error message.

If the user deletes the page mark before renumbering, he should issue the Number command with the "/" range specification, the desired increment, and starting point, if desired.

*N10,/(CR)

Renumber the current page with an

increment and starting point of 10.

After renumbering, the current number and page will be set to the number of the last renumbered line. An increment of 0 will result in the error message "ILLEGAL COMMAND". If the specified increment is so large that it causes some line numbers to be greater than 99999, the high-order digits of these line numbers will be dropped. SOS will then print the error message "%WRAP-AROUND" along with the page on which the problem occurred. The page will be out of order and should be renumbered with a smaller increment.

5.3 The ALTER-COMMAND (INTRALINE-EDIT-MODE) - "A"

The Alter command (referred to by the Find command) is used to make changes within a line without having to retype the line. The Alter command accepts a range specifier indicating those lines to be altered.

*A<range><CR>

For each of the lines in the range, SOS prints the line number, and then enters a special intraline edit mode which has its own commands. A pointer, maintained within the line, points to the next character in the line that will be affected by the next command. The special intraline edit commands do not appear on the user's teletype. Thus, the line shown on the teletype at the end of the intraline edit is exactly the same as it is in the file (except for the backslashes and the characters between them). Any command in this mode may be preceded by a number which will cause the command to be repeated for that number of characters. The line number of the next line in the range will be typed out automatically as soon as intraline editing is completed on the present line.

The current line is determined by the last line edited.

If there are no lines in the specified range, SOS will print the error message "%NO SUCH LINE(S)"

INTRALINE EDIT COMMANDS

SPACE	This will advance the pointer one position to the right. The character that is passed over is printed on the teletype. If the pointer is already at the extreme right of the line, this command is ignored.
RUBOUT	This moves the pointer one character to the left. This does not delete anything as it would ordinarily in input mode. The characters moved over are printed between single backslashes (\XXX\). If the pointer is already at the extreme left of the line, a (CR)<LF> is automatically performed. The line number will be reprinted, and the pointer will again point to the beginning of the line.
CCHARACTER	This causes SOS to accept the next character from the teletype. The character pointed to will be replaced by the new character, and the pointer will be moved one character

to the right. The character accepted from the teletype is printed on the line in place of the one that is deleted. If the pointer is at the extreme right of the line, the command is ignored. During this command, any RUBOUT that is used will be ignored. (<LF>), (<CR>), and \$ will abort the remainder of the command.

D

This will delete the character pointed to. The character(s) deleted will be surrounded by double backslashes (\\XXX\\). The pointer will point to the right of the last character deleted. If the command is preceded by a number, all the characters that were deleted will be printed and enclosed in double backslashes. The command is ignored if the pointer is at the extreme right of the line.

ISTRING \$

This will insert the given strings into the line immediately before the character to which the pointer points. The pointer remains pointing to the character to the right of the inserted string. The characters being inserted are typed out on the teletype.

If RUBOUT is typed before the \$, the last character that is inserted will be deleted. If more RUBOUTs are typed than characters inserted, the characters to the left of the insert are deleted. The characters deleted are surrounded by single backslashes. If enough characters are inserted to make the line exceed 147 characters, the message "%LINE TOO LONG" is printed, and SOS will return the user to command mode without performing any of the alterations specified for that line.

If a <CR> is typed before the \$, the insert will be terminated; intraline edit will terminate as in the <CR> command.

If a (<LF>) is typed before the \$, a (<CR>(<LF>)) is inserted at that point. This creates a new line consisting of any subsequently inserted characters and the rest of the old line to the right of the pointer. The number of the new line is determined as follows:

1. If the I command is preceded by a number, this number acts like a temporary increment and is added to the current line number to obtain the new line number.
2. If the I command is not preceded by a number, then the new line number is determined by adding the current increment to the current line number. The new line is inserted at this number unless it would be out of order.
3. If the new line would be out of order in (2) above, the new line number will be halfway between the current line number and the next line number. If the next line number is only one more than the current line number, an "%OUT OF ORDER" error message will be printed. SOS

will retype the line number and the contents of the line to the left of the pointer. It is then possible to insert more characters.

If no errors occur while making the new line, the pointer will be left pointing to the first character of the new line, and the current line will be set to the new line just created.

<CR>

If there is no more editing to be done, that portion of the line to the right of the pointer will be printed out; intraline editing is completed for that line and command returns to the user, unless there are more lines in the Alter command range to be edited.

E

This has the same effect as the <CR> key, except that the rest of the line is not printed on the teletype.

Q

The Quit command terminates intraline editing without making any changes in the line presently being altered. SOS returns to command mode.

~U

This restarts intraline editing on the same line. The line will be restored to its original state, and intraline editing can be started as soon as the line number is reprinted.

SCHARACTER

The Skip command will accept one character from the user's teletype (without printing it) and will move the pointer to the right until the next occurrence of that character. Every character passed over will be printed. The character that is currently being pointed to also will be printed without being compared to the character specified in the command. If the character does not occur in the rest of the line, the pointer will be moved to the extreme right of the line.

KCHARACTER

The Kill command is the same as the Skip command except that it deletes all the characters passed over instead of printing them. If there are no other occurrences of the specified character in a line, the command will be ignored (rather than the remainder of the line being deleted). As with the D command, the last 3 characters deleted are printed between double backslashes if a number precedes K.

R

The Replace command is the same as a Delete command followed by an Insert (even if a number is included before R).

L

The Line command prints the remainder of the line to the right of the pointer, performs a (CR)(<LF>), prints the number of the line, and continues the intraline editing with the pointer pointing to the first character of the line.

P

This prints the remainder of the line to the right of the

pointer, performs a (CR)(<LF>), prints the line number, and moves the pointer to the right; the characters passed over are printed until the pointer reaches the same position it had before the command was issued.

J This Justify command inserts a (CR)(<LF>) where the pointer is currently pointing, and concatenates the portion of the line to the right of the pointer onto the beginning of the next line. The current line number is set to the new line number.

The error message "%LINE TOO LONG" will be given if the new line that is created contains more than 147 characters. The error message "%NO NEXT LINE" will be printed if this line is the last one on the page. Either of these two errors will cause the J command to be ignored; the line number and the portion of the line to the left of the pointer will be typed out again.

W The Word command moves the pointer to the right, passes over blanks and tabs, then over the first "word" encountered, and then over subsequent blanks and tabs (until the next word). Everything that is passed over is printed.

A "word" is defined as any collection of letters, numbers, and the separators ".", "%", and "\$". The other separators which delimit a word will cause the scan to stop after the word and subsequent blanks and tabs are passed over. (See also Section 6.20 to set separators.)

XSTRING \$ The X command moves the pointer over blanks and tabs, deleting them as it moves, and then deletes the next "word" and the following blanks and tabs. After the deletion, a string of text is inserted to replace the word and subsequent blanks which were deleted.

Any other character, used as a command, will be ignored in intraline edit mode.

5.3.1 The EXTEND-COMMAND - "X" - The Extend command is another way to enter intraline edit mode.

*X<range><CR>

With this command, however, the pointer is moved immediately to the extreme right of the line, the characters passed over are printed on the teletype, and the user is in the intraline insert mode. The desired text is then typed in followed by <CR>.

<text><CR>

If a (<LF>) is typed instead of a <CR>, a return, linefeed occurs at that point. This creates a new line consisting of anything that is subsequently typed in, since the user is still in insert mode. The number of the new line is determined by the methods described in the Section on the ALTER-COMMAND.

Note that a <CR> must be used eventually to end the insertion, unless the sequence of line numbers does not permit another line. In this case, the user will be returned to SOS command level.

If an \$ is typed before the <CR>, the user may then issue any of the intraline edit mode commands.
 <text> \$ <CR>

The following form of the command will suppress the type-out of the original line:

 *X<range>,S<CR>

Only the characters that the user types for the insertion will be printed on the teletype.

5.4 INDIRECT-COMMANDS

If a particular sequence of commands is used frequently, it is possible to insert these commands in a file. Then SOS can be instructed to read commands from the file, rather than from the teletype keyboard. The command

 *(FILENAME)<CR>

will do this, provided the list of commands is found in the named file. When SOS sees this command, it will begin taking its commands from the file. Only the following will be printed out on the user's teletype:

1. Error messages
2. Printout from the P command
3. Items found by the F command
4. Substitutions made by the S command
5. Lines altered by the A command

When SOS has executed the last command in the "indirect command" file, the user will be returned to command mode.

When the user is inserting commands into the file, he may have to enter such characters as \$ or a <CR> without the (<LF>).

6.0 APPENDIX A - SUMMARY OF SOS COMMANDS

alter

*A<range><CR>

Next Character - SPACE
Last Character - RUBOUT
Change - CCHARACTER
Delete - D
Insert - ISTRING \$
End Alter Mode - <CR>
End Alter Mode - E (without printout)
Quit - Q
Start Over - ^U
Skip - SCHARACTER
Kill - KCHARACTER
Replace - R
Line - L
Print - P
Justify - J
Word - W
Delete Word and Insert Text - XSTRING \$

<LF> prints the next line in the file.
*<LF>
00700 BEL: OUTSTR [7]
*

Cline1,line2:line3 copies lines from line2 through line3 to
the lines following line1.
*C200:400:800
*

Dline1:line2 deletes from line1 to line2.
*D400:700
4 LINES (00400/1:00700) DELETED

E saves the file and ends SOS.
*E
[DSKC:STATS.FORE[27,4020]]

ES saves the file without line numbers.
*ES
[DSKC:STATS.FORE[27,4020]]

EQ ends SOS without saving the file.
*EQ
.

Fstring\$line1:line2 prints the first line located between
line1 and line2 which contains the

string of characters.
*FNEWER\$400:10300
002300 NEWER FILES WILL BE SAVED ON
*

G saves the file, ends SOS, and executes the last compile, load, execute, or debug command.
*G
[DSKC:STATS.FORE27,4020]]
the last compil-class command is executed at this point.

Iline1 starts insert mode for line1.
*I500
00500 104 FORMAT (' DIST= ',(F))
00600 \$
*

Nnumber numbers a file starting with number and adding number to produce subsequent line numbers.
*N20
*

Pline1:line2 prints the contents of the file from line1 through line2.
*P100:300
00100 TYPE 101
00200 101 FORMAT (' HELLO.')

00300 END

*

Rline1:line2 deletes from line1 through line2 and then starts inserting at line1.
*R100:200
00100 TYPE 105
00200 105 FORMAT (' GOODBYE.')

2 LINES (00100/1:00200) DELETED

*

Sstring1\$string2\$11:12 finds all occurrences of string1 between lines 11 and 12 and then replaces string1 with the new string 2.
*STHAN\$THEN\$100:800
00700 THEN, HE BETTER GET READY FOR
*

Tline1,line2:line3 copies from line2 through line3 to line1, then deletes from line2 through line3.
*T100,400:500
*

W saves the file, but does not end SOS.
*W
[DSKC:STATS.FORE27,4020]]
*

7.0 APPENDIX B

*** CONVENTIONS AND ABBREVIATIONS USED IN THIS MANUAL ***

* * * * *
GENERAL SYSTEM ABBREVIATIONS
* * *

ADR : an octal address.

DEV: : any physical or logical device name (e.g. DSKB: , LPT: , CDR:).
Colon must be included.

FILE.EXT : name and extension that is associated with a file. Name can be 1 to 6 characters in length and extension can be 1 to 4 characters in length. The first character of extension must always be a period (.). Extension is optional. Example: progrrm.cbl

JOB : name that is assigned to a job that is being entered into or is in one of system queues. This name can be 1 to 6 characters in length.

<NNN> : a 3 digit code indicating protection or access privileges of file. See the PROTECT command for an explanation of the PROTECTION-CODES .

[P,PN] : project and programmer number assigned to each user. 2 numbers that make up P,PN must be separated by a comma or a slash.

SWITCH : one or more switches used to modify command strings.

TEXT : a message to be sent to designated user or terminal.

* * * * *
HARDCOPY AND VIDEO TERMINAL ABBREVIATIONS
* * * * *

<ALT> : indicates key labeled 'ALT' or 'ESC' (ALTMODE or ESCAPE).

<CR> : whenever this symbol appears, it indicates that you should press key labeled 'return' or 'carriage-return'.

 : indicates key that deletes or rubs-out characters.

<ESC> : indicates key labeled 'ALT' or 'ESC' (ALTMODE or ESCAPE).

<LF> : indicates key labeled "line feed".

<RUB> : indicates key that deletes or rubs-out characters. the CPU at the same time.

8.0 INDEX

<ALT>	37
<CR>	37
	37
<ESC>	37
<LF>	37
<NNN>	37
<RUB>	37
ABBREVIATIONS	37
ADR	37
ALTER-COMMAND	29
BACKUP-FILE	19
CONVENTIONS	37
COPY-COMMAND	15, 26
CTRL-R	6
CTRL-R.	6
CTRL-U	6
DELETE-COMMAND	10, 22
DEV:	37
DISK-FILE,	4
EDIT-MODE-COMMANDS	21
END-AND-QUIT-COMMAND	18
END-AND-STRIP-COMMAND	17
END-COMMAND	17
Erasing-characters	6
ESCAPE	7
ESCAPE-KEY.	7, 9
EXTEND-COMMAND	32
FILE.EXT	37
FIND-COMMAND	12, 25
GO-COMMAND	17
INC1=ORDER	16
INDEX	38
INDIRECT-COMMANDS	33
INPUT:	4
INSERT-COMMAND	10, 23
INTRALINE-EDIT-MODE	29
JOB	37
JOIN-COMMAND	28
LINE-NUMBERS	5, 20
MARK-COMMAND	21
NO SUCH LINES	22

NUMBER-COMMAND	11, 28
ORDER	16
Out-of-order	27-28
PAGES	20
PRINT-COMMAND	8
PROTECT	37
PROTECTION-CODES	37
QUIT-COMMAND	18
REPLACE-COMMAND	11, 24
RUBOUT	6
SEARCH-FAILS	13-14
SUBSTITUTE-COMMAND	13
SWITCH	37
TEXT	37
TEXT-EDITOR.	3
TRANSFER-COMMAND	15
WORLD-COMMAND	18
WRAP-AROUND	12
Wrap-around	27, 29
[P,PNI]	37

9.0 *** READER COMMENT FORM ***

Use this page to forward comments about this manual.

Did you find errors in this manual? If so, specify errors and page numbers.

Did you find the manual understandable, usable and organized? Can you make any suggestions to help other users?

Is there sufficient documentation on the features/topics that you utilize?

Which sections would you recommend be expanded? Which shortened? Which rewritten?

Please indicate the type of user/reader you represent...

- ☐ student
 - ☐ full-time
 - ☐ part-time
 - ☐ day
 - ☐ evening
- ☐ BC3 staff employee
- ☐ BC3 faculty employee
- ☐ not associated with the college
- ☐ experienced with computers
- ☐ some prior exposure to computers
- ☐ no prior experience with computers

Your name:-----course/dept:-----

THE HISTORY OF BUCKS COUNTY

THE HISTORY OF BUCKS COUNTY
FROM 1639 TO 1800
BY JAMES H. HARRIS

THE HISTORY OF BUCKS COUNTY
FROM 1800 TO 1850
BY JAMES H. HARRIS

THE HISTORY OF BUCKS COUNTY
FROM 1850 TO 1900
BY JAMES H. HARRIS

THE HISTORY OF BUCKS COUNTY
FROM 1900 TO 1950
BY JAMES H. HARRIS

THE HISTORY OF BUCKS COUNTY
FROM 1950 TO 2000
BY JAMES H. HARRIS

THE HISTORY OF BUCKS COUNTY
FROM 2000 TO THE PRESENT
BY JAMES H. HARRIS

THE HISTORY OF BUCKS COUNTY
FROM 1639 TO THE PRESENT
BY JAMES H. HARRIS

